Athens University of Economics and Business

MSc in Business Analytics

Machine Learning & Content Analytics - PT

Students: Eleni Kakagianni, Elena Styliara, Aris Petrou

Academic student IDs: p2821905, p2821927, p2821918

Quarter: Spring quarter 2019-2020

Instructor: Haris Papageorgiou

Assistant Professors: George Perrakis, Aris Fergadis

October 2020

Table of Contents

Introduction

In context of assignment on behalf of United Nations tech department , our team undertake the project of SDG classification for the creation of their online library. The goal is the implementation of a neural network model which will accomplish the SDG classification of any scientific paper in their library efficiently. United Nations need is to categorize the scientific documents according to their SDGs and in real time to classify each paper that will be retrieved. The Sustainable Development Goals (SDGs), also known as the Global Goals, were adopted by all United Nations Member States in 2015 as a universal call to action to end poverty, protect the planet and ensure that all people enjoy peace and prosperity by 2030. Specifically, the SGD' s topic analysis we will focus on, is a machine learning technique that interprets and categorizes large collections of texts according to individual topics or themes. For this purpose, instead of having to read thousands of text documents to identify the main topic that authors are talking about, a topic analysis tool will be used to do it in seconds.

The first requirement of building the model was to collect a sufficient number of documents from different SDGs. To implement that, we procced with the annotation process where we collect the data-papers and generate our dataset. The site source we use, is the *ScienceDirect* (https://www.sciencedirect.com/) where we searched for scientific, technical, and environmental related publications regarding SDG13. Our scope was to collect 150 papers in total, processed with the annotation method. Our analysis consists of six SDGs and the total number of texts gathered, was 899.

9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
13 CLIMATE ACTION
3 GOOD HEALTH AND WELL-BEING
7 AFFORDABLE AND CLEAN ENERGY
13 CLIMATE ACTION
5 GENDER EQUALITY
10 REDUCED INEQUALITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
17 PARTNERSHIPS FOR THE GOALS

The next step after the data collection is the cleansing of the dataset. The dataset contains three columns ('title', 'abstract', 'initial text'). The text classification methods which we implemented were the following: Multilayer Perceptions (MLPs), Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for different variations. The aim of the model is to recognize targeted words given the title, the abstract and the initial text of any paper respectively and to find the proper SDG group in which belongs to. In addition to the above, we used the CNN and RNN method to construct a model with two inputs and work with a different neural network system architecture.

Finally, we compared the three text classification methods mentioned above based on their metric results and ended up with CNN as the most suitable method.

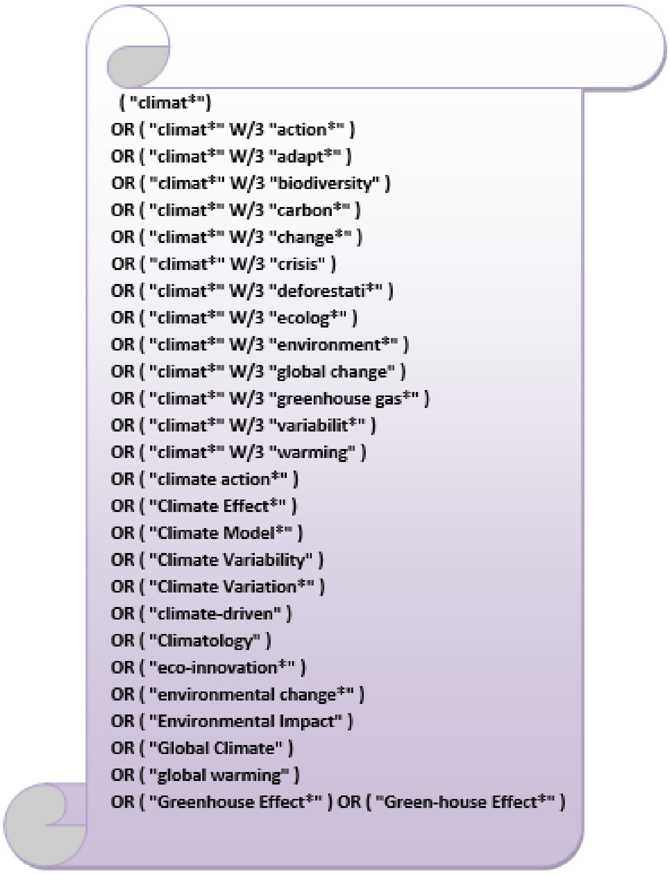
Annotation Process

To begin with, the first and foremost step in the analysis refers to the dataset and the annotation process we used to collect the data-papers and generate our dataset. The most demanding part is the appropriate data collection to train the models and get valuable insights.

Our team gathered scientific papers concerning SDG13. Sustainable Development Goal 13 aims to “take urgent action to combat climate change and its impact”, while acknowledging that the United Nations Framework Convention on Climate Change is the primary international, intergovernmental forum for negotiating the global response to climate change.

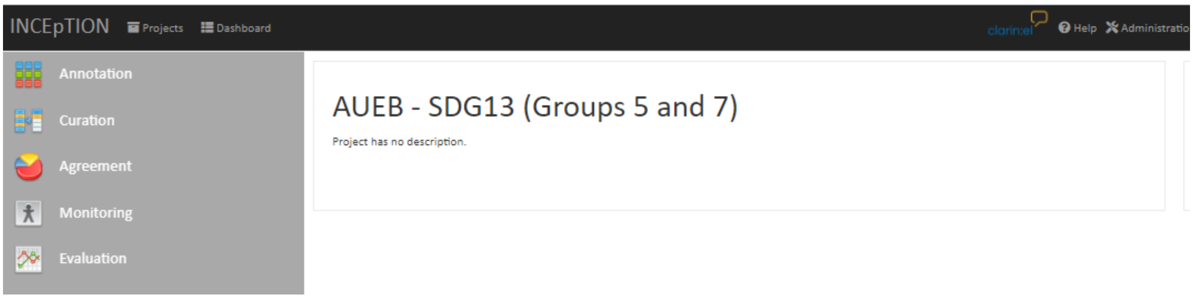
‘’Targets of SDG 13 focus on the integration of climate change measures into national policies, the improvement of education, awareness-raising and institutional capacity on climate change mitigation, adaptation, impact reduction and early warnings. Everyone is needed to reach these ambitious targets. The creativity, knowhow technology and financial resources from all of society is necessary to achieve the SDGs in every context.’’

ScienceDirect is the source site we searched for publications. Specifically, we applied the following query [Figure 1] which includes environmental targeted indicators, into ScienceDirect’s searching tab to specify more suitable papers.



```
 ( "climat*")
OR ( "climat*" W/3 "action*" )
OR ( "climat*" W/3 "adapt*" )
OR ( "climat*" W/3 "biodiversity" )
OR ( "climat*" W/3 "carbon*" )
OR ( "climat*" W/3 "change*" )
OR ( "climat*" W/3 "crisis" )
OR ( "climat*" W/3 "deforestati*" )
OR ( "climat*" W/3 "ecolog*" )
OR ( "climat*" W/3 "environment*" )
OR ( "climat*" W/3 "global change" )
OR ( "climat*" W/3 "greenhouse gas*" )
OR ( "climat*" W/3 "variabilit*" )
OR ( "climat*" W/3 "warming" )
OR ( "climate action*" )
OR ( "Climate Effect*" )
OR ( "Climate Model*" )
OR ( "Climate Variability" )
OR ( "Climate Variation*" )
OR ( "climate-driven" )
OR ( "Climatology" )
OR ( "eco-innovation*" )
OR ( "environmental change*" )
OR ( "Environmental Impact" )
OR ( "Global Climate" )
OR ( "global warming" )
OR ( "Greenhouse Effect*" ) OR ( "Green-house Effect*" )
```

Figure 1: Environmental Indicators Query



AUEB - SDG13 (Groups 5 and 7)

Project has no description.

Figure 2: INCEpTION platform

The annotation process focuses on the annotation of three main components in every document. Namely these components are the Discourse Topic, Argument and Research Theme.

Topic Annotation

The topic is what the discourse is all about and captures the aim or the purpose of the paper. The topic must be indicative of the discourse and not to generic.

Argument Annotation

To find out the material in a text, we need to identify and evaluate its arguments. An argument consists of one or more claims that something is or should be the case, and the evidence which is the justification for why claim or claims should be accepted. The Argument layer has two labels which are the Claim and the Evidence.

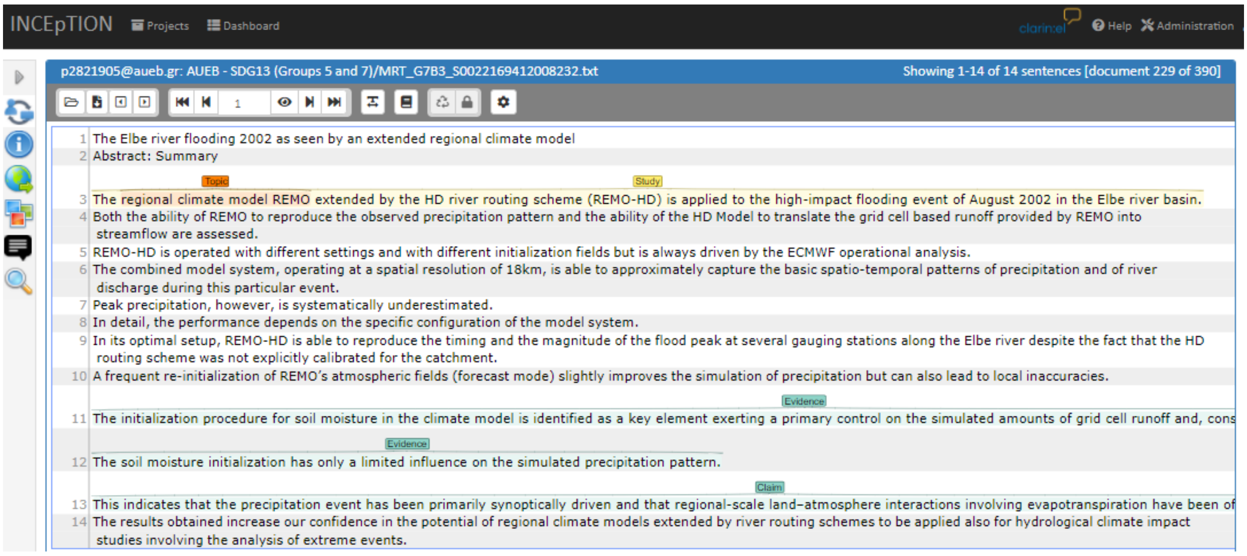
Research Theme Annotation

The Research Theme expresses the overarching goals of the authors and we investigate how do they achieve them. Their goal may be a test (e.g., of a device or a diagnostic tool), a comparison/evaluation (e.g., of methods or treatments), an implementation (e.g., of a novel system). Research theme focus on the goal of the paper on the following aspects, which also are the labels of this layer:

- Clinical Trial
- Device
- Diagnostic Tool
- Drug
- Infrastructure
- Material
- Prototype
- Study
- Treatment
- Other

The label for the Research Theme layer should be only one of the above.

In the following figure [Figure 3], a fully annotated document is presented based on its main components mentioned above.



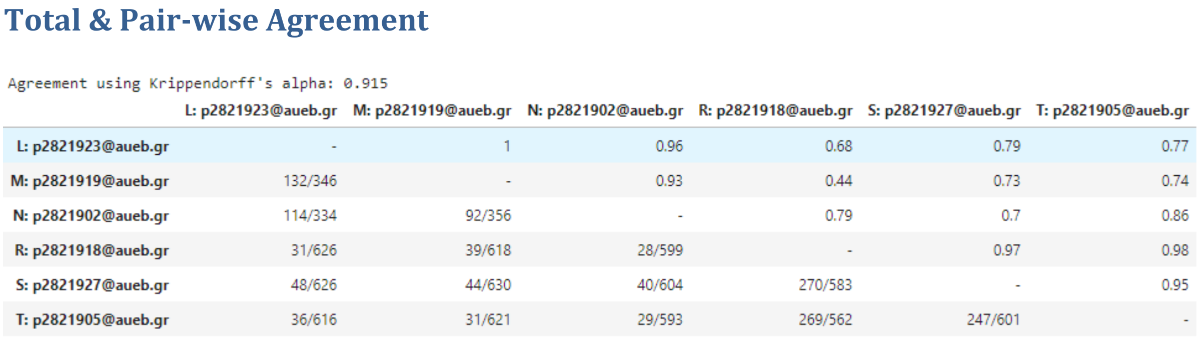
p2821905@aueb.gr: AUEB - SDG13 (Groups 5 and 7)/MRT_G7B3_S0022169412008232.txt

Showing 1-14 of 14 sentences [document 229 of 390]

1 The Elbe river flooding 2002 as seen by an extended regional climate model
2 Abstract: Summary
3 The regional climate model REMO extended by the HD river routing scheme (REMO-HD) is applied to the high-impact flooding event of August 2002 in the Elbe river basin.
4 Both the ability of REMO to reproduce the observed precipitation pattern and the ability of the HD Model to translate the grid cell based runoff provided by REMO into streamflow are assessed.
5 REMO-HD is operated with different settings and with different initialization fields but is always driven by the ECMWF operational analysis.
6 The combined model system, operating at a spatial resolution of 18km, is able to approximately capture the basic spatio-temporal patterns of precipitation and of river discharge during this particular event.
7 Peak precipitation, however, is systematically underestimated.
8 In detail, the performance depends on the specific configuration of the model system.
9 In its optimal setup, REMO-HD is able to reproduce the timing and the magnitude of the flood peak at several gauging stations along the Elbe river despite the fact that the HD routing scheme was not explicitly calibrated for the catchment.
10 A frequent re-initialization of REMO's atmospheric fields (forecast mode) slightly improves the simulation of precipitation but can also lead to local inaccuracies.
11 The initialization procedure for soil moisture in the climate model is identified as a key element exerting a primary control on the simulated amounts of grid cell runoff and, cons
12 The soil moisture initialization has only a limited influence on the simulated precipitation pattern.
13 This indicates that the precipitation event has been primarily synoptically driven and that regional-scale land–atmosphere interactions involving evapotranspiration have been of
14 The results obtained increase our confidence in the potential of regional climate models extended by river routing schemes to be applied also for hydrological climate impact studies involving the analysis of extreme events.

Figure 3: An annotated document including all the components (Title, Research Theme, Claim-Evidence)

We implemented the aforementioned process in all the 150 documents. As soon as each batch has been completed, a detailed report to evaluate our agreement scores and other metrics has been extracted. In the figure below [Figure 4], we can distinguish the final pair-wise agreement scores among all the annotators. Particularly, in the following table the upper half is the pair-wise annotator agreement. The lower half has a pair of number for each pair of annotators. The first number is the number of the common annotated sentences and the second one the total sentences annotated by both annotators (intersection / union).



Total & Pair-wise Agreement

Agreement using Krippendorff's alpha: 0.915

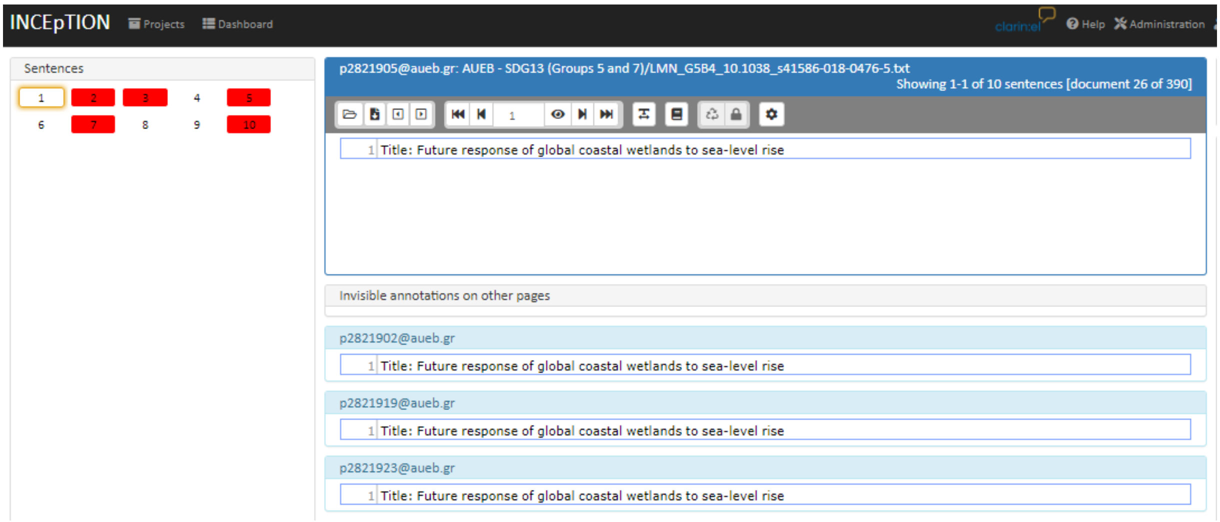
	L: p2821923@aueb.gr	M: p2821919@aueb.gr	N: p2821902@aueb.gr	R: p2821918@aueb.gr	S: p2821927@aueb.gr	T: p2821905@aueb.gr
L: p2821923@aueb.gr	-	1	0.96	0.68	0.79	0.77
M: p2821919@aueb.gr	132/346	-	0.93	0.44	0.73	0.74
N: p2821902@aueb.gr	114/334	92/356	-	0.79	0.7	0.86
R: p2821918@aueb.gr	31/626	39/618	28/599	-	0.97	0.98
S: p2821927@aueb.gr	48/626	44/630	40/604	270/583	-	0.95
T: p2821905@aueb.gr	36/616	31/621	29/593	269/562	247/601	-

Figure 4: Total & Pair-wise Agreement Scores

Curation process

In case of high percentage of disagreement score, only one annotator named curator undertakes to handle those documents through the curation process. Specifically, the curator could see all the annotated sentences in the INCEpTION platform and select the final sentence which is most appropriate based on them perception.

An example of the curation process is presented in the following figure [Figure 5].



p2821905@aueb.gr: AUEB - SDG13 (Groups 5 and 7)/LMN_G5B4_10.1038_s41586-018-0476-5.txt

Showing 1-1 of 10 sentences [document 26 of 390]

1 Title: Future response of global coastal wetlands to sea-level rise

p2821902@aueb.gr

1 Title: Future response of global coastal wetlands to sea-level rise

p2821919@aueb.gr

1 Title: Future response of global coastal wetlands to sea-level rise

p2821923@aueb.gr

1 Title: Future response of global coastal wetlands to sea-level rise

Figure 5: A curation process example in the INCEpTION

Since the annotation process has ended, all the documents of all the SDG groups have been collected to generate the final dataset in which we implement machine learning algorithms and train our models using MLP, CNN and RNN methods.

Data Reprocessing – Text Classification Methods

On this chapter, we present the stage of data reprocessing and the machine learning methods we used to train our models. The results and the selection of the best method will be made on the next chapter.

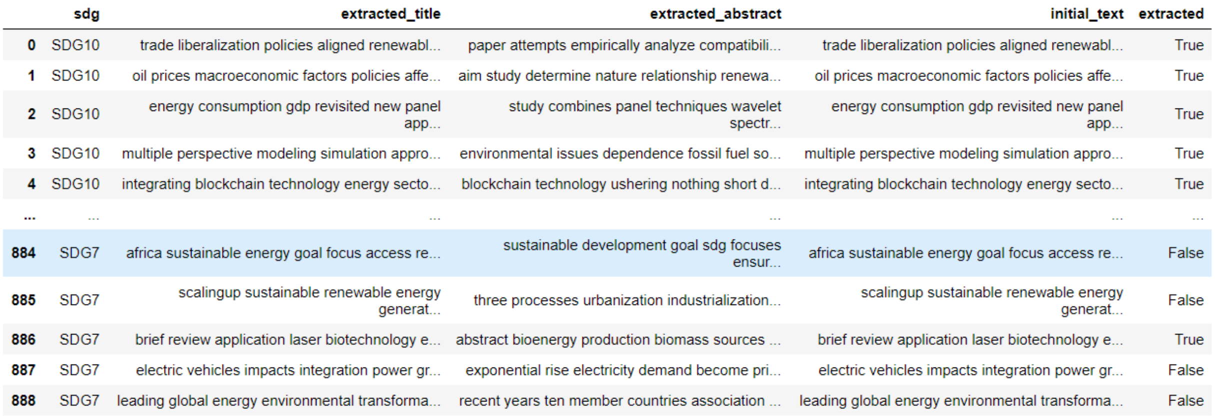
Data Reprocessing

After data collection, the information should be reprocessed and transformed into a more readable/editable format. At first, we imported a list of English stop words in order to validate them against the words of our data frame rows and finally, remove them. Moreover, we count the number of appearances per word in order to note frequently-used words that do not help us classify the text (e.g. 'also', 'using' etc.). We decided, also, not to remove the rows with 'extracted' value: false, since their information was not corrupted and could be contributed to the classification.

The next step, is to perform the following actions:

- Remove the word 'Title:'
- Convert all the word letters to lowercase
- Remove all the special characters
- Remove all the words that are also part of the stopwords list
- Remove frequently-used words that are not 'stopwords'

The final format of our dataset is as follows in the [Figure 6]:



	sdg	extracted_title	extracted_abstract	initial_text	extracted
0	SDG10	trade liberalization policies aligned renewabl...	paper attempts empirically analyze compatibili...	trade liberalization policies aligned renewabl...	True
1	SDG10	oil prices macroeconomic factors policies affe...	aim study determine nature relationship renewa...	oil prices macroeconomic factors policies affe...	True
2	SDG10	energy consumption gdp revisited new panel app...	study combines panel techniques wavelet spectr...	energy consumption gdp revisited new panel app...	True
3	SDG10	multiple perspective modeling simulation appro...	environmental issues dependence fossil fuel so...	multiple perspective modeling simulation appro...	True
4	SDG10	integrating blockchain technology energy secto...	blockchain technology ushering nothing short d...	integrating blockchain technology energy secto...	True
...	...	...	...	...	...
884	SDG7	africa sustainable energy goal focus access re...	sustainable development goal sdg focuses ensur...	africa sustainable energy goal focus access re...	False
885	SDG7	scalingup sustainable renewable energy generat...	three processes urbanization industrialization...	scalingup sustainable renewable energy generat...	False
886	SDG7	brief review application laser biotechnology e...	abstract bioenergy production biomass sources ...	brief review application laser biotechnology e...	True
887	SDG7	electric vehicles impacts integration power gr...	exponential rise electricity demand become pri...	electric vehicles impacts integration power gr...	False
888	SDG7	leading global energy environmental transforma...	recent years ten member countries association ...	leading global energy environmental transforma...	False

Figure 6: The final format of the dataset

Finally, we create a csv file (**sdg_classification_data.csv**) with the above reprocessed dataset to use it while applying different text classification methods. Our final dataset contains information about 889 documents. The input variables are the following: ***extracted_title***, ***extracted_abstract***, ***initial_text*** and the output: ***sdg***.

In the figure below [Figure 7] is presented the number of existing documents per SDG.

SDG	Number of documents
SDG3	264
SDG5	247
SDG13	243
SDG10	50
SDG12	44
SDG7	41

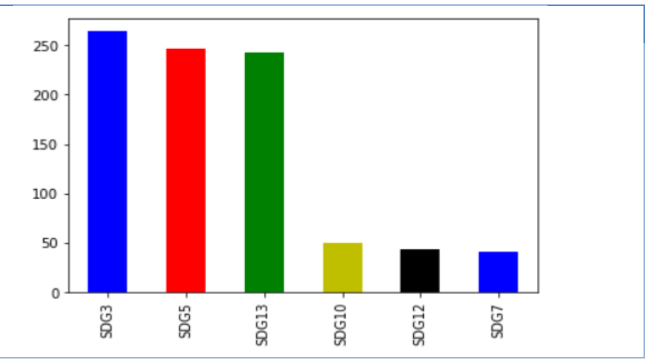


Figure 7: Number of documents per SDG

In addition, some information about the most common words of the documents can de noticed below [Figure 8]:

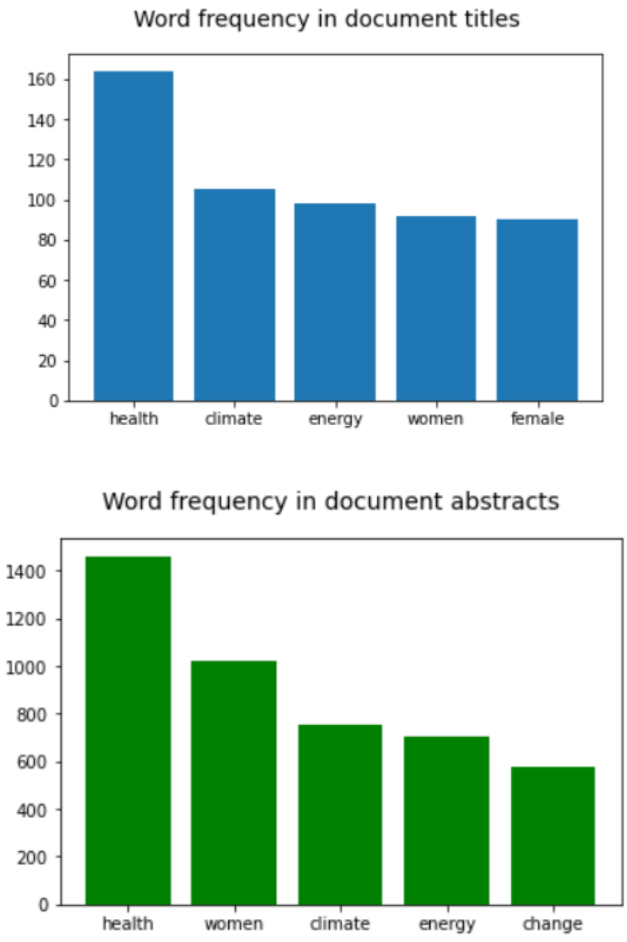


Figure 8: The most common words of the document Titles & Abstracts

The most common words in the document titles are the following: ***health, climate, energy, women, female*** while in the abstracts: ***health, women, climate, energy, change***.

Text classification neural network methods

For the SDG classification problem, we worked with 3 classed of artificial neural networks. We used 4 different inputs: title, abstract, initial text (title & abstract) and 2-input. For the final methods evaluation, we used the **blind dataset** shared by Mr. Fergadis. That dataset includes 99 documents, in which we apply the same preprocessing methods with the initial dataset.

The artificial neural network classes we implemented are the following:

- Multilayer Perceptrons (MLPs)
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs)

Multilayer Perceptron (MLP)

Multilayer Perceptrons, or MLPs for short, are the classical type of neural network. They are comprised of one or more layers of neurons. Data is fed to the input layer, there may be one or more hidden layers providing levels of abstraction, and predictions are made on the output layer, also called the visible layer.

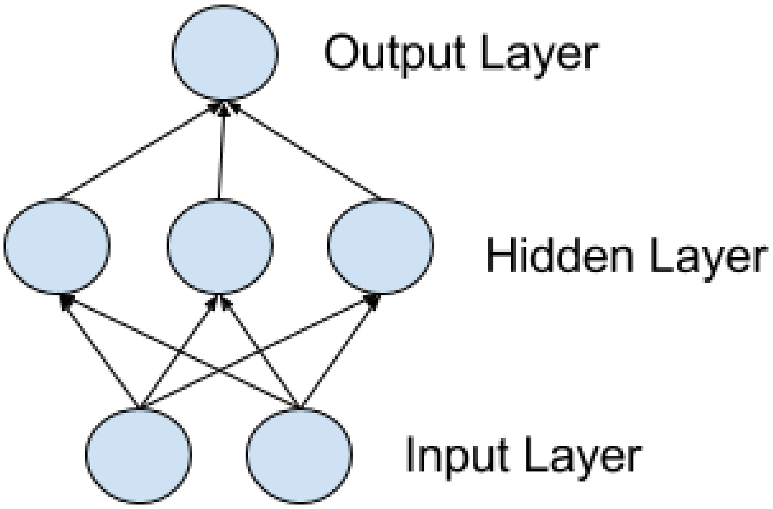


Figure 9: Classical type of Neural Network(NN)

At this case, we have a feed forward network model where the information moves in only one direction—forward—from the input nodes, through the hidden nodes and to the output nodes. There are no cycles or loops in the network. In the [Figure 10] a feed forward network model is presented.

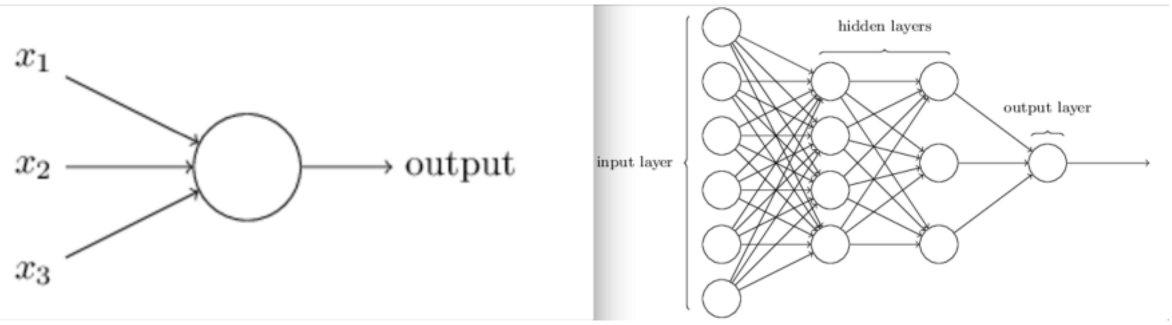


Figure 10: Architecture of a Feed Forward Network model

This neural network requires numerical data. Since we had categorical data, we used *One Hot Encoding* and *CountVectorizer* to convert them into real – valued(numerical) representation. The first method has been used in the output value ('sdg'), while the second in the input values ('extracted_title', 'extracted_abstract', 'initial_text'). Finally, we end up with the normalization of both input and output variables; thus, the variables have been rescaled into 0 and 1 values.

In parallel with the above process, we split the dataset in train and test datasets and the train dataset in train and validation datasets. Thus, we worked with 3 different datasets. The next step, was to define the model parameters. We defined the following parameters:

- **max_words**: the max words per input (max_words = 300)
- **nb_classes**: number of classes for the output labels (nb_classes = 6)
- **nb_epochs**: number of epochs that we will train our feed forward network (30 epochs)
- **batch_size**: the batch size of the data that will be fed to the model while training (batch_size = 32)
- **dropout_rate**: how many neurons should be shut down per epoch (dropout_rate = 0.4)

After defining the model parameters, we compiled the model, setting the parameters as below:

- **loss**: “categorical_crossentropy”, since we have categorical output values
- **optimizer**: “adam”, Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments.
- **metrics**: “accuracy”

Then we build the model, creating dense layers and ‘relu’ activations on the outputs.

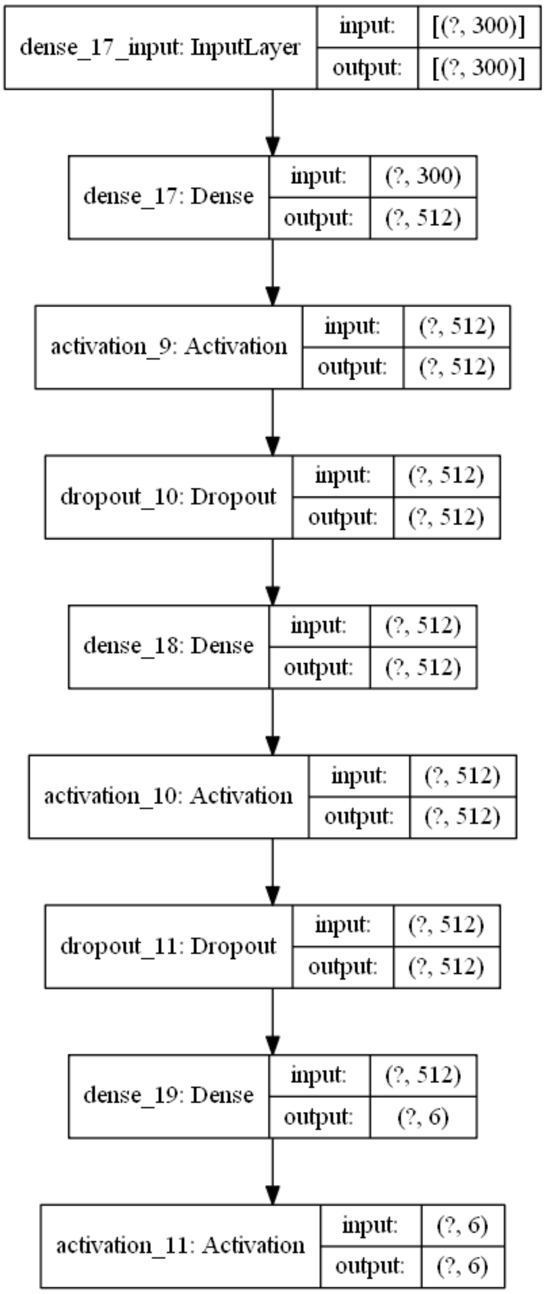


Figure 11: Model plot for Title input

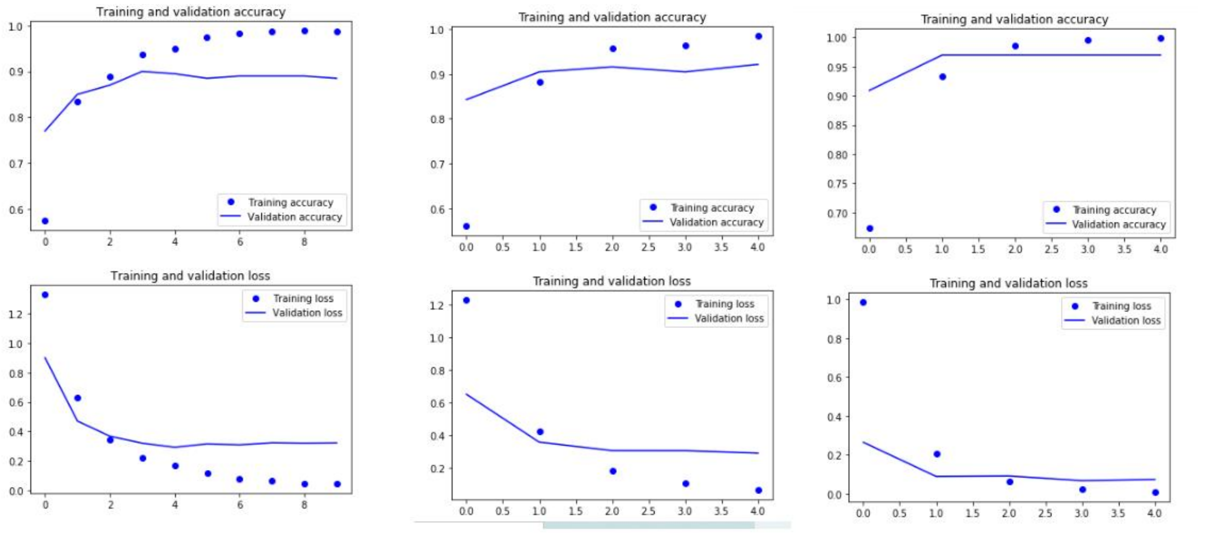
Model Evaluation on the Test Set

Finally, we trained the model per input and gained the following accuracies to see how it performs [Figure 12]:

Input	Accuracy (%)
extracted_title	91.09
extracted_abstract	94.95
initial_text	91.57

Figure 12: Accuracy metrics per input

Accuracy and loss plots between training and validation data for MLP.



Accuracy and loss plots-title *Accuracy and loss plots-abstract* *Accuracy and loss plots-initial text*

Figure 13: Accuracy and loss plots between training and validation data

Finally, it is noticeable that the validation loss and validation accuracy diagrams are not in sync with the training loss and training accuracy. It is clear in all the variations that the overfitting phenomenon is present and further parameter tuning is required.

Convolutional Neural Network (CNN)

Convolutional Neural Networks use the concept of a "convolution", a sliding window or "filter" that passes over the image, identifying important features and analyzing them one at a time, then reducing them down to their essential characteristics, and repeating the process.

The architecture we used while implementing CNN, starts with an input sentence broken up into word embeddings. Words are broken up into features and are fed into a convolutional layer and then into a hidden layer. The results of the convolution are "pooled" to a representative number. This number is fed to a fully connected neural structure, which makes a classification decision based on the weights assigned to each feature within the text. An architecture sample can be found below [Figure 14]:

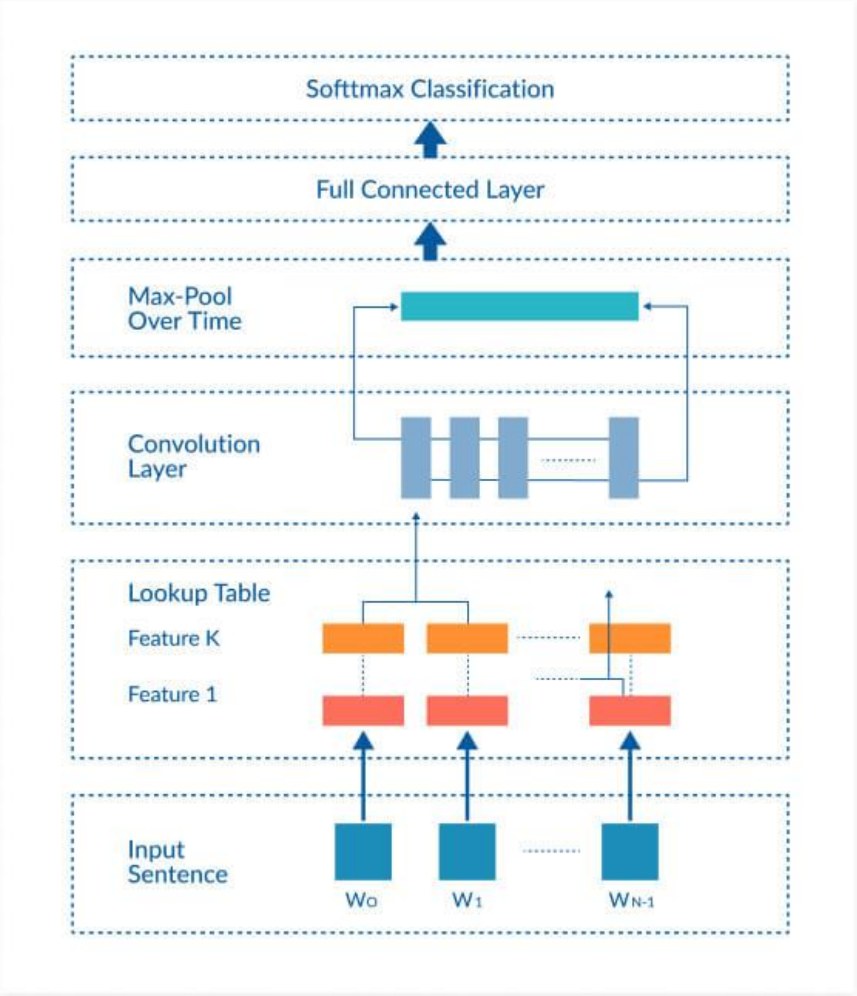


Figure 14: CNN Architecture

It is highly important to point out that we train 2 different models: 1 without including Dropout Rate and 1 including Dropout Rate. **Dropout** hyperparameter is the probability of training a given node in a layer, where 1.0 means no **dropout**, and 0.0 means no outputs from the layer. We set the dropout = 0.2.

The model parameters we defined are the following:

- **embedding_dims**: word embedding dimension (embedding_dims = 50)
- **nof_filters**: number of filters (nof_filters = 25)
- **batch_size**: the batch size of the data that will be fed to the model while training (batch_size = 64)
- **kernel_size**: (width)*(height) of the filter mask (kernel_size = 3)
- **hidden_dims**: the number of neurons in the hidden layer (hidden_dims = 50)

After defining the model parameters, we compiled the model, setting the parameters as below:

- **loss**: “categorical_crossentropy”, since we have categorical output values
- **optimizer**: “adam”, Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments.
- **metrics**: “accuracy”

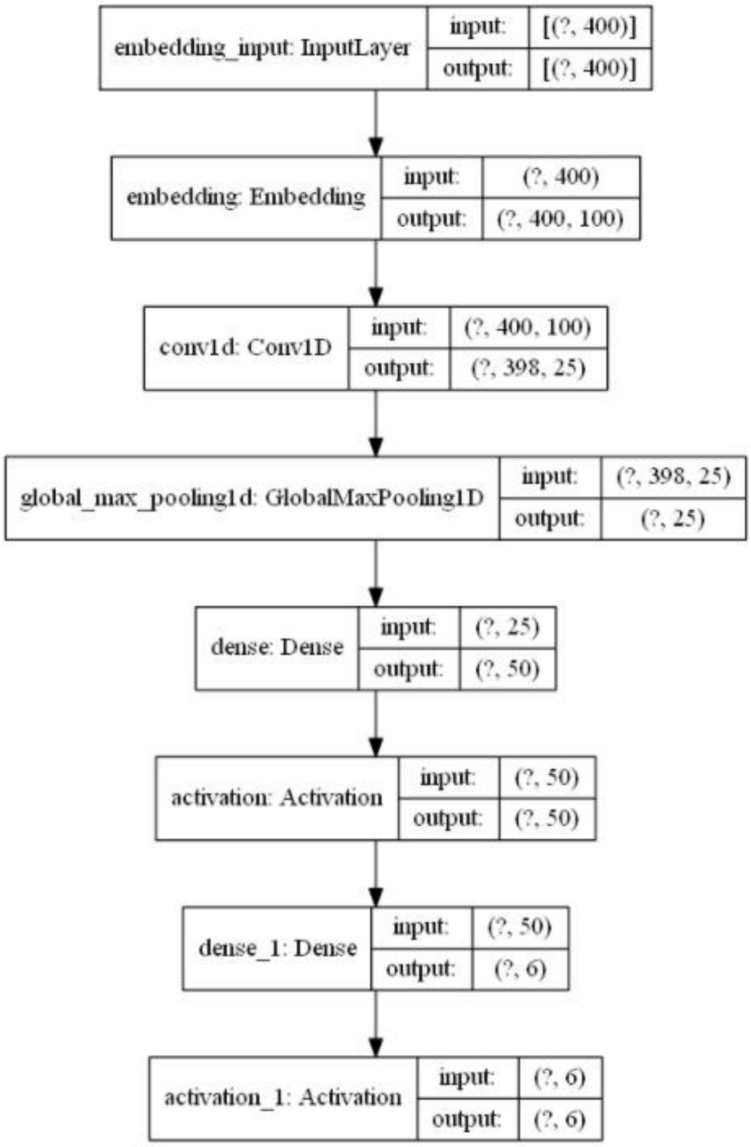


Figure 15: Model plot for Title input

]

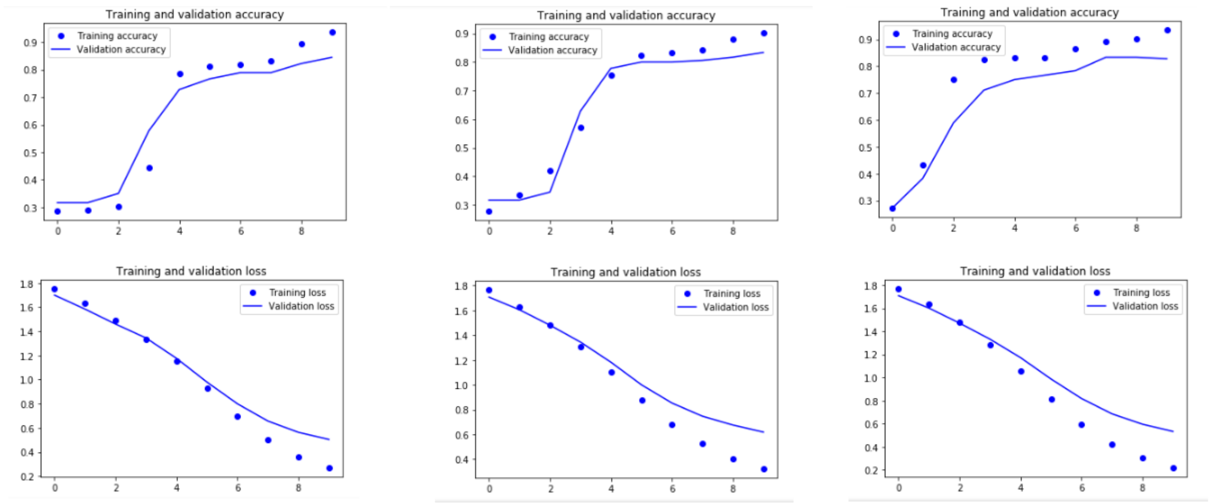
Model Evaluation on the Test Set

Finally, we trained the model per input and gained the following accuracies in order to see how it performs [Figure 16]:

Input	Accuracy (%)
extracted_title	91.92
extracted_abstract	88.90
initial_text	92.93

Figure 16: Accuracy metrics per input

Accuracy and loss plots between training and validation data for CNN.



Accuracy and loss plots-title *Accuracy and loss plots-abstract* *Accuracy and loss plots-initial text*

Figure 17: Accuracy and loss plots between training and validation data

Finally, it is noticeable that the validation loss and validation accuracy both are in sync with the training loss and training accuracy. Even though the validation loss and accuracy line are not linear, our model is not over fitting: the validation loss is decreasing and not increasing, and there is not much gap between training and validation accuracy. Therefore, it is obvious that our model's generalization capability became much better since the loss on both test set and validation set was only slightly more compared to the training loss.

Recurrent neural network (RNN)

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. Recurrent Neural Networks are a type of Neural Network, where the output from previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus, RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

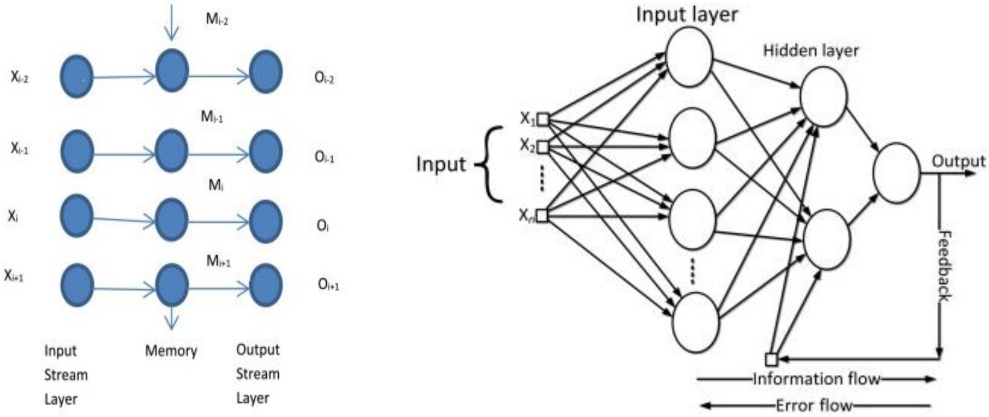


Figure 18: RNN Model

RNN have a “memory” which remembers all the information about what has been calculated. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks.
Moreover, it is important to point out that we used LSTM architecture. Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. LSTM is well-suited to classify process and predict time series given time lags of unknown duration. It trains the model by using back-propagation.
Another noticeable thing to mention is that we trained two different models: one without including Dropout Rate and one with Dropout Rate. **Dropout** hyperparameter is the probability of training a given node in a layer, where 1.0 means no **dropout**, and 0.0 means no outputs from the layer. We set the dropout = 0.2.

The defined model parameters are the following:

- **MAX_NB_WORDS (=50000)**
- **embedding_dims**: word embedding dimension (embedding_dims = 100)
- **batch_size**: the batch size of the data that will be fed to the model while training (batch_size = 64)
- **dropout_rate**: how many neurons should be shut down per epoch (dropout_rate = 0.2)

After defining the model parameters, we compiled the model, setting the parameters as below:

- **loss**: “categorical_crossentropy”, since we have categorical output values
- **optimizer**: “adam”, Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments.
- **metrics**: “accuracy”

We built the model, creating dense layers and ‘relu’ activations on the outputs [Figure 19]:

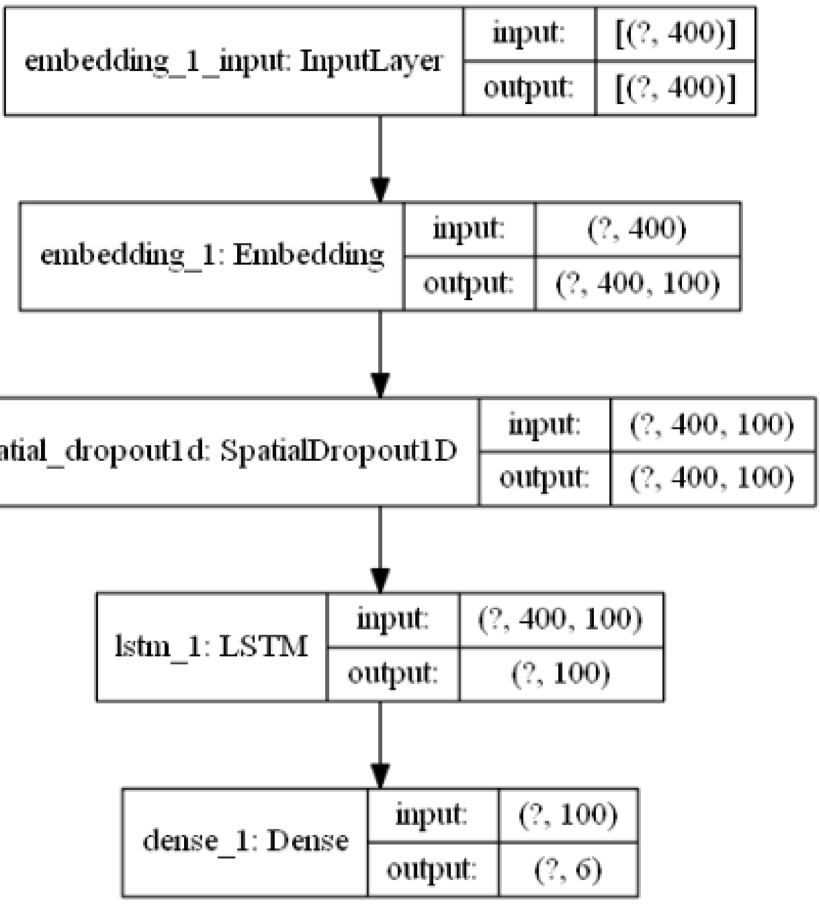


Figure 19: Model plot presentation

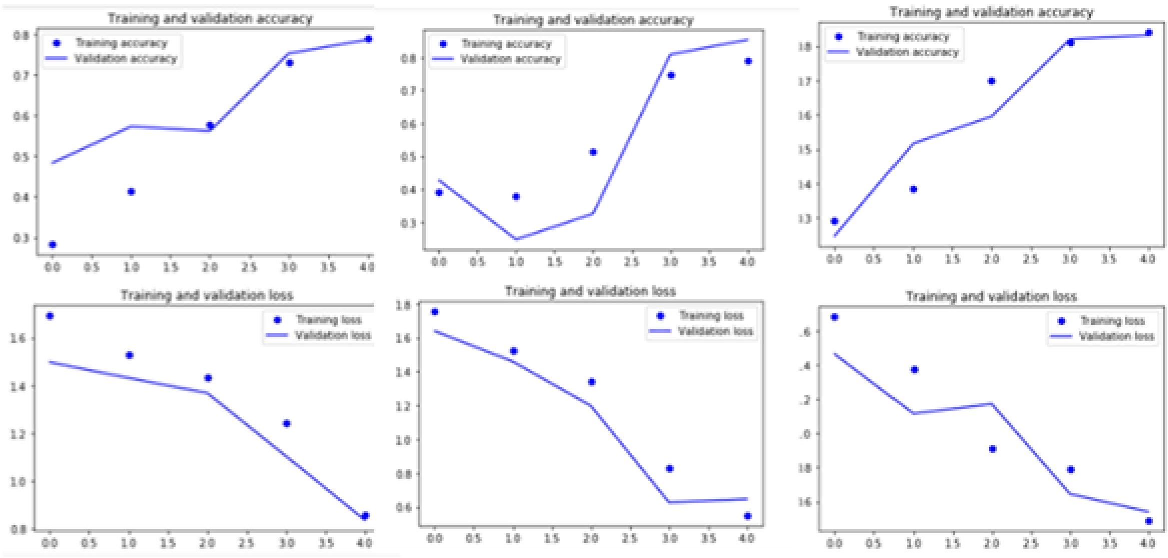
Model Evaluation on the Test Set

Finally, we trained the model per input and gained the following accuracies to see how it performs [Figure 20]:

Input	Accuracy (%)
extracted_title	82.82
extracted_abstract	86.87
initial_text	87.88

Figure 20: Accuracy metrics per input

Accuracy and loss plots between training and validation data for RNN.



Accuracy and loss plots-title *Accuracy and loss plots-abstract* *Accuracy and loss plots-initial text*

Figure 21: Accuracy and loss plots between training and validation data

Finally, it is noticeable that the validation loss and validation accuracy both are in sync with the training loss and training accuracy. Even though the validation loss and accuracy line are not linear, our model is not over fitting: the validation loss is decreasing and not increasing, and there is not much gap between training and validation accuracy. Therefore, it is obvious that our model's generalization capability became much better since the loss on both test set and validation set was only slightly more compared to the training loss.

Multi-input Neural Network for Text Classification

An alternative method for training artificial neural networks is the multi input text classification. We implemented two inputs text classification in CNN and RNN method, respectively. The input variables which we used are the ‘extracted_title’ and the ‘extracted_abstract’. An indicative architecture diagram from CNN method is the following [Figure 22]:

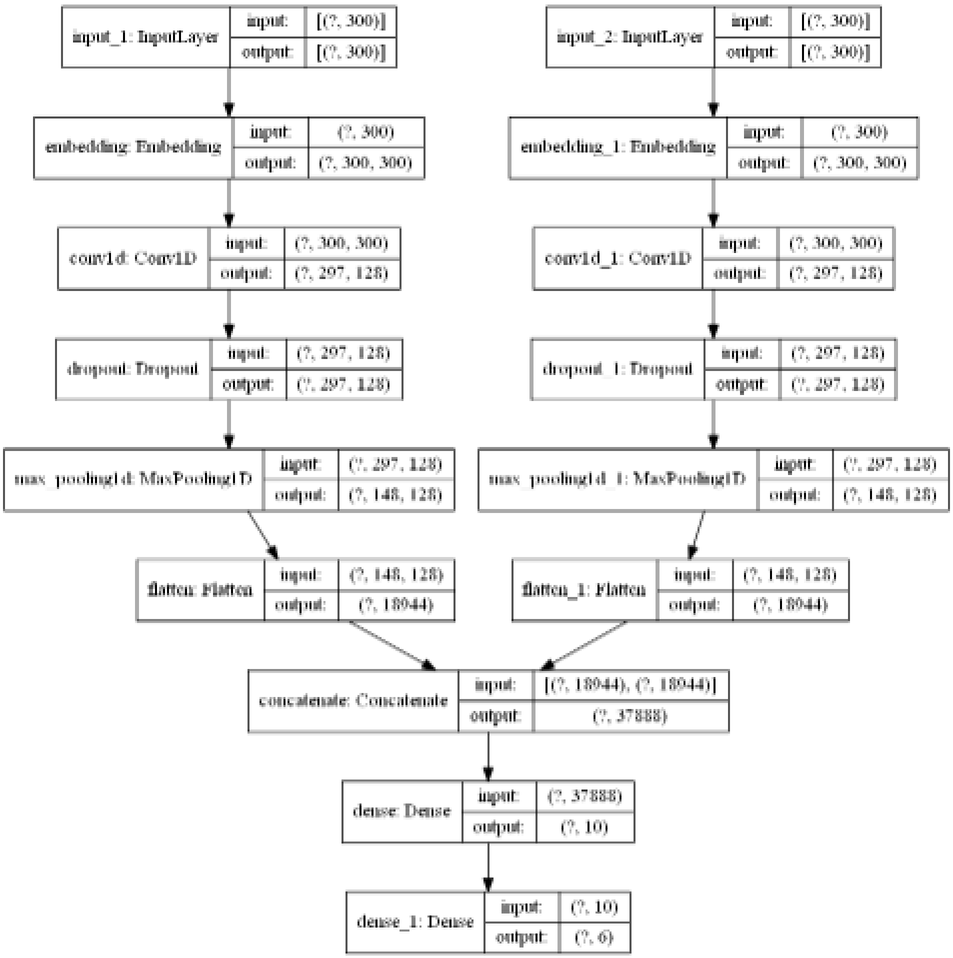


Figure 22: CNN Architecture Diagram

The parameters of both two inputs are presenting below:

- **embedding_dims**: word embedding dimension (embedding_dims = 50)
- **nof_filters**: number of filters (nof_filters = 25)
- **batch_size**: the batch size of the data that will be fed to the model while training (batch_size = 64)
- **kernel_size**: (width)*(height) of the filter mask (kernel_size = 3)
- **hidden_dims**: the number of neurons in the hidden layer (hidden_dims = 50)

After defining the model parameters, we compiled the model, setting the parameters as below:

- **loss**: “categorical_crossentropy”, since we have categorical output values
- **optimizer**: “adam”, Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments.
- **metrics**: “accuracy”

However, the metric results were not as expected the accuracy achieved, did not exceed 55%, even after sufficient parameters tuning. It is preferable to use one input architecture, thus we select the parameter initial text which is the concatenation of the other two parameters.

Results

On this chapter, we select the best neural model according to the metric's results such as the accuracy and loss. In addition, the confusion matrices and the classification reports created via Jupyter notebook, are presented.

Neural Network method selection

The selection of the most suitable neural network method takes into account many different parameters. At first, we checked the accuracy results that are presented below [Figure 23].

	MLP	CNN	RNN
Avg accuracy	95.62	91.25	85.86
Avg loss	0.124	0.3166	0.5137

Figure 23: Avg Accuracy and Avg Loss results of MLP, CNN & RNN

Based on the above results, the obvious selection would be the MLP method. However, after taking a close look at the plots of MLP method, we noticed that the gap between training and validation accuracy in MLP method is really large. This gap is a clear indication of overfitting, since the larger the gap, the higher the overfitting.

Sometimes visualizing the training loss vs. validation loss or training accuracy vs. validation accuracy over a number of epochs is a good way to determine if the model has been sufficiently trained. In our case, this visualization leads us to reject the MLP method and select the method with the second results, the CNN.

Evaluation Metrics

Roc Curve

AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve, and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at corresponding the documents to the right SDGs or not. The ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.

In the following figure [Figure 24], CNN Roc Curves of abstract, title and initial text are presented. Observing the roc curves below, we notice that 100% of the SDG3 documents are classified correctly.

CNN Abstract Roc Curve **CNN Title Roc Curve** **CNN Initial Text Roc Curve**

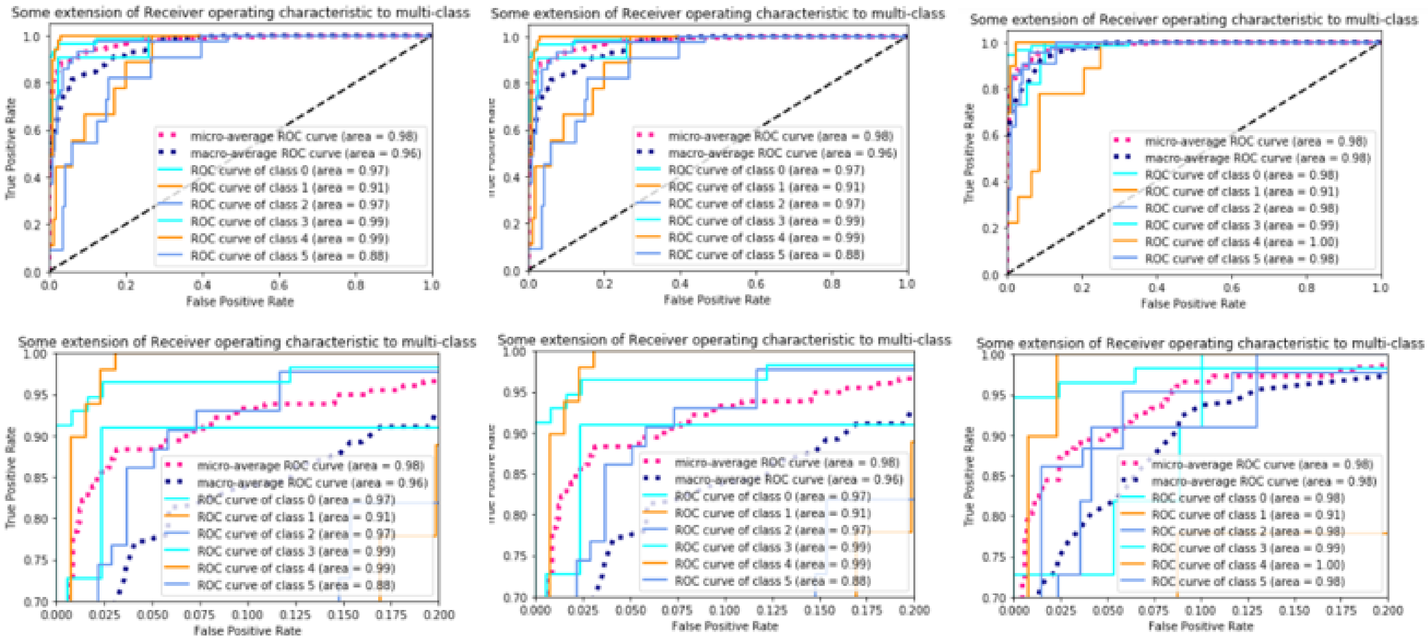
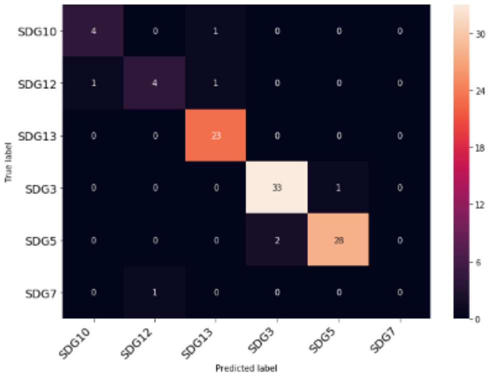


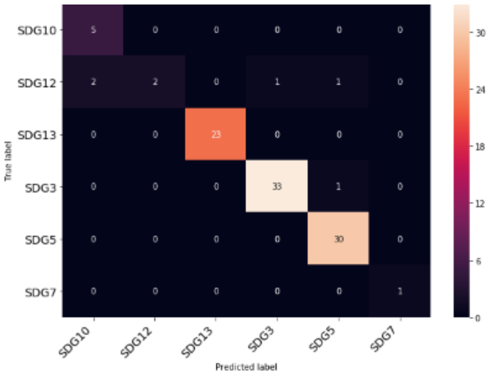
Figure 24: CNN Roc Curves of the Abstract, Title & the Initial Text

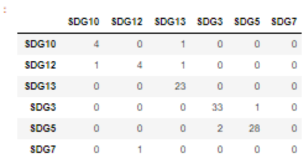
Confusion Matrix

The confusion matrix is a tabular way of visualizing the performance of the prediction model. Each entry in a confusion matrix denotes the number of predictions made by the model where it classified the classes correctly or incorrectly for multi-class machine learning models.

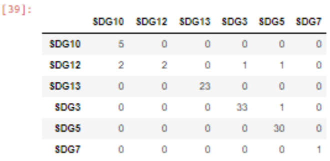
In the figure below [Figure 25], we present all the confusion matrices of the abstract, title and the initial text in the CNN method. The prediction label and true labels show us which prediction class we are dealing with. The matrix diagonal represents locations in the matrix where the prediction and the truth are the same, so this is where we want the heat map to be darker.
In the figures below we observe that the most predictions have classified correctly and there are minimal observations which have classified wrong.



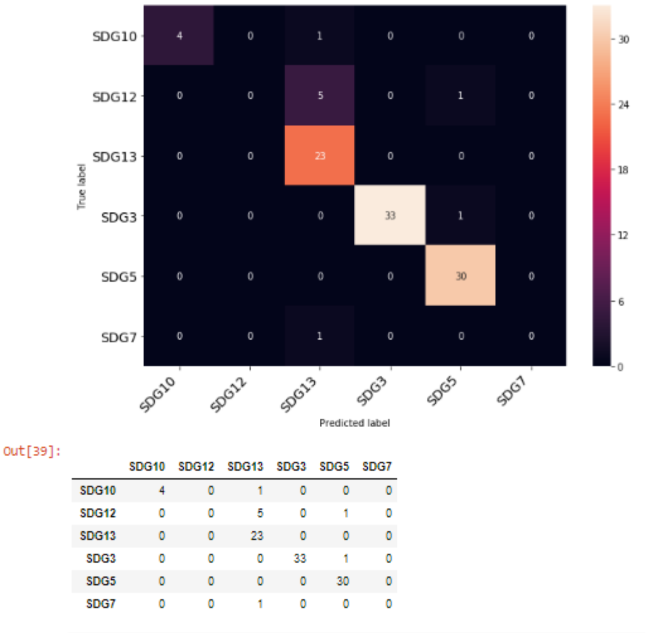



:

	SDG10	SDG12	SDG13	SDG3	SDG5	SDG7
SDG10	4	0	1	0	0	0
SDG12	1	4	1	0	0	0
SDG13	0	0	23	0	0	0
SDG3	0	0	0	33	1	0
SDG5	0	0	0	2	28	0
SDG7	0	1	0	0	0	0


[39]:

	SDG10	SDG12	SDG13	SDG3	SDG5	SDG7
SDG10	5	0	0	0	0	0
SDG12	2	2	0	1	1	0
SDG13	0	0	23	0	0	0
SDG3	0	0	0	33	1	0
SDG5	0	0	0	0	30	0
SDG7	0	0	0	0	0	1



Out[39]:

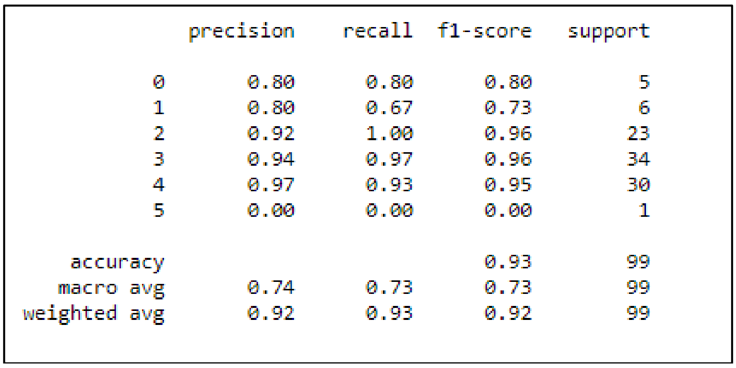
	SDG10	SDG12	SDG13	SDG3	SDG5	SDG7
SDG10	4	0	1	0	0	0
SDG12	0	0	5	0	1	0
SDG13	0	0	23	0	0	0
SDG3	0	0	0	33	1	0
SDG5	0	0	0	0	30	0
SDG7	0	0	1	0	0	0

Figure 25: Confusion Matrix for CNN Title, Abstract & Initial Text

We observe that in the confusion matrix of the 'initial text', SDG12 is not appeared in the diagonal of the matrix whereas in the two other matrices this is not happening. The reason about this phenomenon probably is that the class of SDG12 has only 5 documents and the possibilities of a bad prediction are quite high.

Classification Report

Classification report will help us in identifying the misclassified classes in more detail manner. Specifically, we are able to observe for which class the model performed bad out of the given six classes [Figure 26].



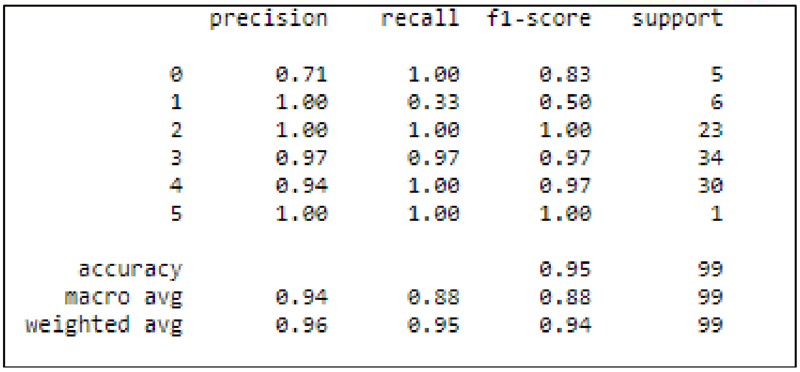
```
              precision    recall  f1-score   support

           0       0.80      0.80      0.80         5
           1       0.80      0.67      0.73         6
           2       0.92      1.00      0.96        23
           3       0.94      0.97      0.96        34
           4       0.97      0.93      0.95        30
           5       0.00      0.00      0.00         1

    accuracy                           0.93        99
   macro avg       0.74      0.73      0.73        99
weighted avg       0.92      0.93      0.92        99
```

Classification Report CNN_title

In the figure above we present the classification report about CNN for the title variable. The classifier is under-performing for class 5 regarding both precision and recall. This happens due to the lack of documents of class 0 (only 1 document).

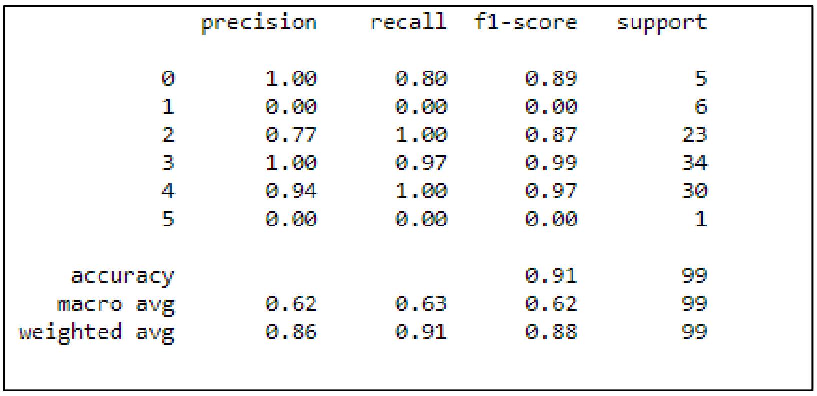


```
              precision    recall  f1-score   support

           0       0.71      1.00      0.83         5
           1       1.00      0.33      0.50         6
           2       1.00      1.00      1.00        23
           3       0.97      0.97      0.97        34
           4       0.94      1.00      0.97        30
           5       1.00      1.00      1.00         1

    accuracy                           0.95        99
   macro avg       0.94      0.88      0.88        99
weighted avg       0.96      0.95      0.94        99
```

Classification Report CNN_abstract



```
              precision    recall  f1-score   support

           0       1.00      0.80      0.89         5
           1       0.00      0.00      0.00         6
           2       0.77      1.00      0.87        23
           3       1.00      0.97      0.99        34
           4       0.94      1.00      0.97        30
           5       0.00      0.00      0.00         1

    accuracy                           0.91        99
   macro avg       0.62      0.63      0.62        99
weighted avg       0.86      0.91      0.88        99
```

Classification Report CNN_initial_text

Figure 26: Classification Report of CNN Title, Abstract & Initial Text

In the figure above we present the classification report about CNN for the initial text variable. The classifier is under-performing for class 1 and 5 regarding both precision and recall. This happens due to the lack of documents of class 0 (only 1 document) and class 5 (only 5 documents).

Predictions

The final part of our project, after selecting the best method, is to make predictions on the blind dataset, shared by Mr. Fergadis. On this purpose, we predicted the titles, the abstracts and the initial tests using the CNN method and created a new csv document with our predictions. Our predictions refer to the percentage of matched classes by the three above variables and contain a detailed report for the different variations of our dataset.

You may find below the statistic results along with the output plots [Figure 27], [Figure 28]:

	Titles	Titles_matched (%)	Abstracts	Abstracts_matched (%)	Initial Texts	Texts_matched (%)
Matched	91	91.91	86	86.87	90	90.9
Not matched	8	8.09	13	13.13	9	9.1

Figure 27: Matched and Unmatched Titles, Abstracts & Initial Texts

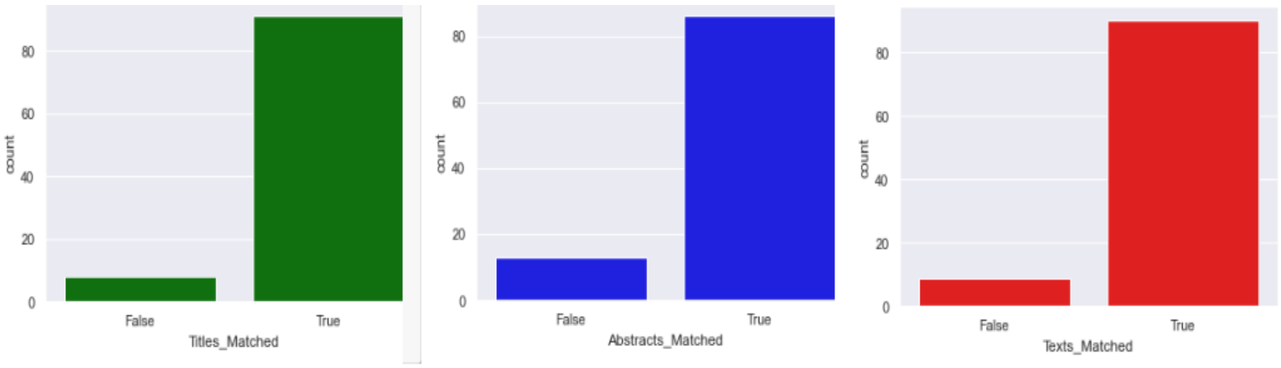
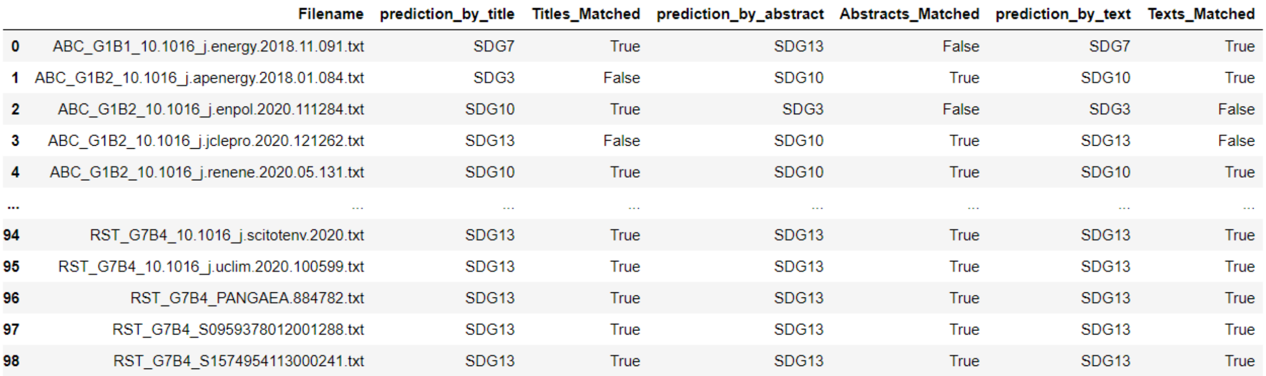


Figure 28: Bar-plots of Matched and Unmatched Titles, Abstracts & Initial Texts

The final dataset containing the predictions, has the following format [Figure 29]:



	Filename	prediction_by_title	Titles_Matched	prediction_by_abstract	Abstracts_Matched	prediction_by_text	Texts_Matched
0	ABC_G1B1_10.1016_j.energy.2018.11.091.txt	SDG7	True	SDG13	False	SDG7	True
1	ABC_G1B2_10.1016_j.apenergy.2018.01.084.txt	SDG3	False	SDG10	True	SDG10	True
2	ABC_G1B2_10.1016_j.enpol.2020.111284.txt	SDG10	True	SDG3	False	SDG3	False
3	ABC_G1B2_10.1016_j.jclepro.2020.121262.txt	SDG13	False	SDG10	True	SDG13	False
4	ABC_G1B2_10.1016_j.renene.2020.05.131.txt	SDG10	True	SDG10	True	SDG10	True
...	...	...	...	...	...	...	...
94	RST_G7B4_10.1016_j.scitotenv.2020.txt	SDG13	True	SDG13	True	SDG13	True
95	RST_G7B4_10.1016_j.uclim.2020.100599.txt	SDG13	True	SDG13	True	SDG13	True
96	RST_G7B4_PANGAEA.884782.txt	SDG13	True	SDG13	True	SDG13	True
97	RST_G7B4_S0959378012001288.txt	SDG13	True	SDG13	True	SDG13	True
98	RST_G7B4_S1574954113000241.txt	SDG13	True	SDG13	True	SDG13	True

Figure 29: The final dataset with predictions

Conclusions

The final model is trained with CNN method and considering the small number of the available texts (almost 1000), achieved precise results(more than 90%). The delivered tool will reduce the time required for SDG's categorization and by extension, the time for the online library implementation. The online library will assist each country's representatives to enrich their knowledge of their assignment targets. Moreover, it is important to mention that an alternative use of this tool in other projects could be accomplished with a small configuration of the model parameters.

Project Implementation

We used Anaconda Jupyter notebook environment, locally in our machines, to implement the classification project for all the aforementioned methods(CNN, RNN, MLP). Python version used was 3.8.3. We needed about 20 minutes to run and fit our models per method.

Team Members

Our team consists of 3 members. Namely:
Eleni Kakagianni
Aris Petrou
Eleni Styliara

The respective emails are:
p2821905@aueb.gr
p2821918@aueb.gr
p2821927@aueb.gr

Time Planning

The initial time plan we had projected was as follows [Figure 30]:

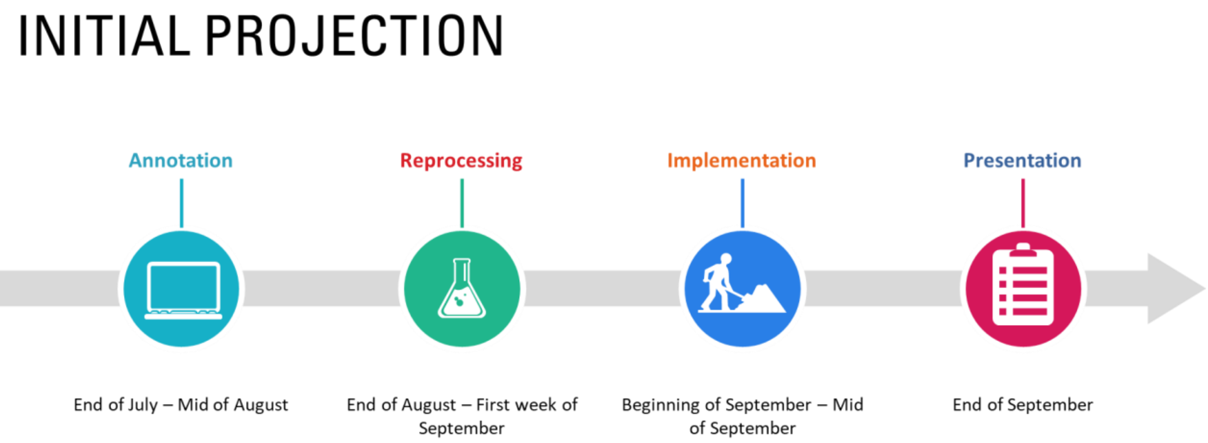

INITIAL PROJECTION
Annotation
Reprocessing
Implementation
Presentation
End of July – Mid of August
End of August – First week of September
Beginning of September – Mid of September
End of September

Figure 30: Initial Projection Plan

However, due to delays in the annotation process, every part of the projection plan delayed for about 2 weeks and had been formed as follows [Figure 31]:

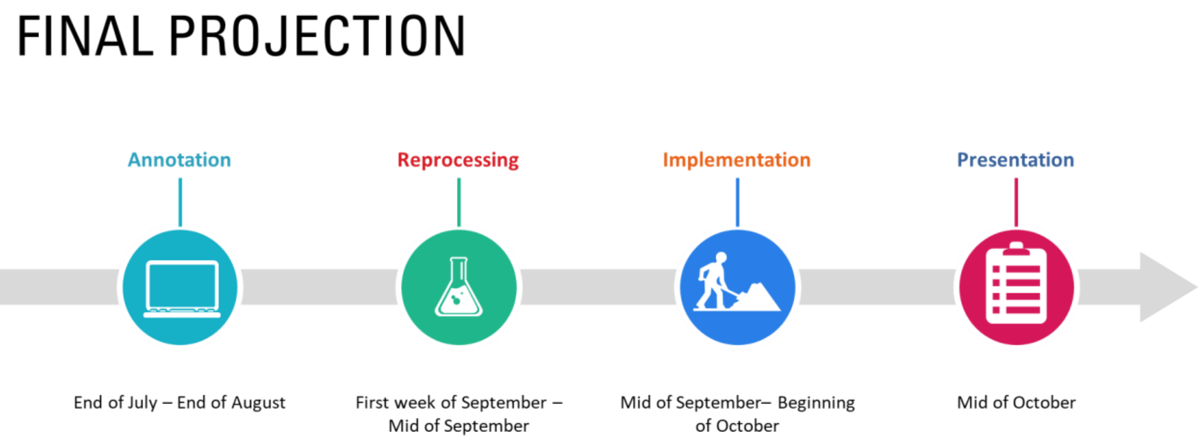

FINAL PROJECTION
Annotation
Reprocessing
Implementation
Presentation
End of July – End of August
First week of September – Mid of September
Mid of September– Beginning of October
Mid of October

Figure 31: Final Projection Plan

Bibliography

✓ Course Labs (1-5)
✓ https://towardsdatascience.com/
✓ https://www.datacamp.com/
✓ https://www.wikipedia.org/
✓ https://missinglink.ai/

/**\